

PREPARED BY:

Diogo Magalhães, Licenciatura
Eng.Informática

PREPARED FOR:

ESTG - Programação Ambiente Web



P.PORTO

August 11, 2019

PandaPanda SOLUTIONS

PROJECT **ManageMyApartment**



Relatorio ManageMyApartment

Apresentacao Projeto	2
Apresentacao Staff	3
Sumario Projeto	4
Cronograma	4
Software Utilizado	5
Requisitos Negocio	7
Representacoes Visuais do Desenvolvimento da App	8
Modulos Importantes Utilizados	13
Conceitos Relevantes	14
Conclusoes Finais	15
Documentacao Extra	16

Sobre ManageMyApartment

O projeto “ManageMyApartment” refere uma empresa ficticia presente no Enunciado, cuja funcao, se trata de replicar uma situacao em contexto real de uma plataforma informatica com o objetivo de simplificar e melhorar o processo de gestao dos servicos prestados por essa entidade comercial bem como oferecer capacidade de administracao e logistica de clientes e o processo de publicacao de alugueres de apartamentos, bem como as suas ofertas (“bookings”).

Para tal efeito, foram arquitetadas tres classes para o formar a aplicacao web de forma a satizfazer o modelo de negocio evidenciado no enunciado.

Tendo em conta a natureza e dinamica das funcoes dos clientes nesta aplicacao web, onde, apos registo, podem tanto prestar anuncios/artigos de alugar de apartamentos bem como gerar ofertas a apartamentos doutros utilizadores da plataforma foi decidido elaborar uma classe “user” onde esta podera ter tanto acesso a funcoes de administrador a residencias de sua autoria bem com funcoes de cliente para as restantes disponiveis.

Uma vez que o projeto é constituído apenas por um elemento, naturalmente a esse mesmo membro fora inputado todo processo de planeamento e concecao do produto final.

PandaPanda Team



Diogo Magalhaes, Aluno Licenciatura Engenharia Informatica na ESTG

Ola, sou o Diogo e sou entusiasta por Informatica e por Pandas!

Como nao tenho experiencia com ursos peludos monocromaticos, vou partilhas as minhas experiencias no ambito da informatica.

Devido a ter participado em varios projetos tanto desenvolvidos em linguagens nao orientadas a objects (como C++), tanto como o oposto (Java e JavaScript), julgo ter adquirido um conhecimento agraçante e ecletico das linguagens de programacao de alto nivel da atualidade.

Surge agora uma nova oportunidade de embarcar num novo desafio de desenvolver uma Aplicacao em Ambiente Web utilizando framework Node.js. Desta forma pretento melhor compreender os paradigmas de programacao orientada a eventos de forma assincrona e obter maior capacidade de agilizar esforco computacional do lado do cliente para os servidores. Desta forma sera possivel criar aplicacoes de alta escalabilidade.

Vamos a isso!

Sumario Plano do Projecto

ManageMyApartment

O projeto “ManageMyApartment” refere uma empresa ficticia presente no Enunciado, cuja funcao, se trata de replicar uma situacao em contexto real de uma plataforma informatica com o objetivo de simplificar e melhorar o processo de gestao dos servicos prestados por essa entidade comercial bem como oferecer capacidade de administracao e logistica de clientes e o processo de publicacao de alugueres de apartamentos, bem como as suas ofertas (“bookings”).

Project Schedule

Requirements	Criar Prototipo	Especificacoes	Desenvolvimento	Estabilizacao	Deploy
5 Dias	2 Dias	2 Dias	10 Dias	2 Dias	3 Dias

ManagMyApartment Project Planing

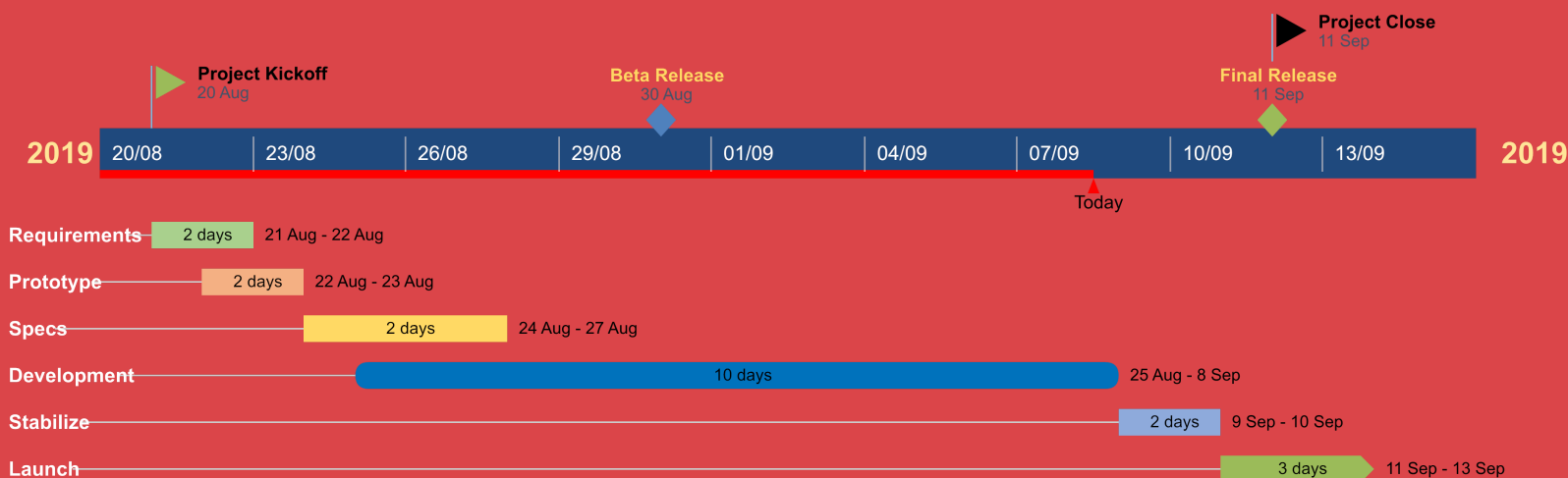


Imagem1 - Diagrama de Gantt

Software utilizado

ManageMyApartment



Pug foi o *middleman* elegido para o Node.js. Este *template engine* permite injetar dados e produzir HTML. Basicamente, em run time, Pug substitui as variaveis no nosso ficheiro com valores reais, que depois envia o resultado final HTML string ao client.

Bootstrap foi utilizado para invocar no header um template de design escolhido para embelezar o HTML gerado pelo Pug para css.



Git foi a ferramenta utilizada para gerir o historial do codigo fonte, aliado do servico de alojamento de repositorios do **GitHub**.



Software utilizado

ManageMyApartment



Visual Studio Code (v1.37) foi a minha ferramenta de eleicao uma vez que, apesar de ser apenas um editor de codigo, este pode ser utilizado como um IDE installando plugins que estao disponiveis no marketplace.

Express.js é uma framework baseada em Node.js para construir aplicacoes web. Uma vez que este é baseado no modulo middleware Noje.js significa que por vez usa desfruta do http module. Com ele iremos respeitar a arquitetura MVC (Arquitetura Modelo-Visão-Controlador).

express



MongoDB é um gestor de Bases de Dados orientada a documentos open source. Embora este seja seja um “Non-relational database” que representa dados em colecoes de documentos JSON, a estrategia que elegi para o desenvolvimento desta App foi de Referencias (o que e uma caracteristica de Bases de dados Relacionais, como por exemplo o “Mysql”).

Para melhor visualizacao dos dados e compreensao da estrutura da Base de Dados, utilizei o MongoDB Compass que e a Interface Grafica de Utilizador oficial do MongoDB.

Requisitos de Modelo de Negócio

ManageMyApartment

Munido ferramentas e das especificações do enunciado fornecido no Moodle de Programação em Ambiente Web de Época Especial 2018-19 começamos interpretar e arquitetar com grande detalhe os seguintes pontos:

-Modelo de negócio da empresa ManageMyApartment:

A aplicação web pretende que seja feita uma gestão de contas de clientes bem como dos seus anúncios e ofertas de alugueres de apartamentos.

-Arquitetura padrão MVC inicialmente no projeto criamos 3 folders(models, routes, views). Para isso foram criadas três folders iniciais de Modelos de dados para "User", "Article" e "Booking", com respetivos Controladores que por sua vez invocam Views para interagir com o utilizador.

Seguidamente foram criados folder "config" onde irá ser armazenado dois javascript files. Um para fazer conexão do nosso web api com o MongoDB através da instalação do módulo mongoose e outra para a implementação do middleware de autenticação Passport do Node.js com JSON web Token (JWT).

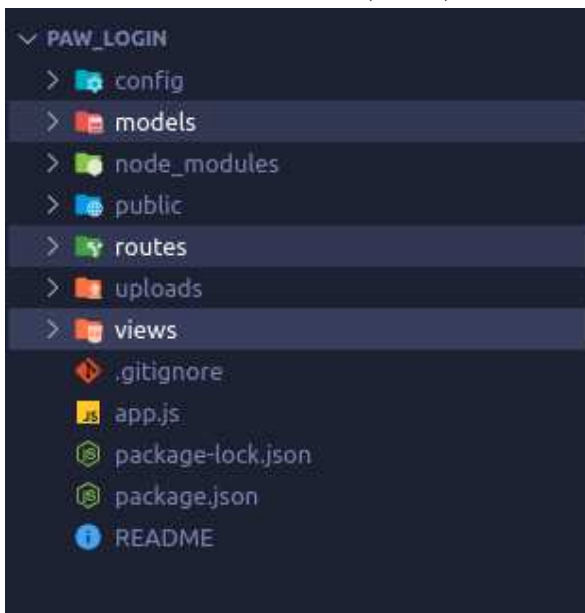
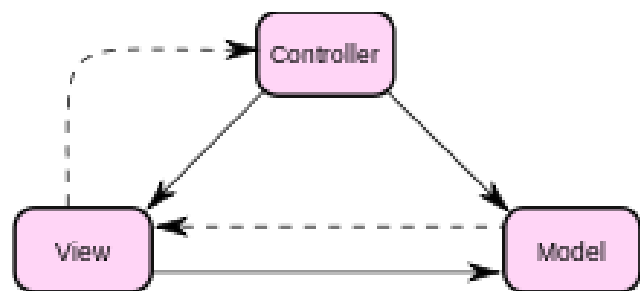


Imagem 2.2 - Como foi estruturado o padrão MVC neste projeto



Um diagrama simples exemplificando a relação entre *Model*, *View* e *Controller*. As linhas sólidas indicam associação direta e as tracejadas indicam associação indireta.

Imagem 2.1 - Diagrama Simples MVC

Fonte: <https://pt.wikipedia.org/wiki/MVC>

Requisitos de Modelo de Negócio

ManageMyApartment

Requisitos Funcionais

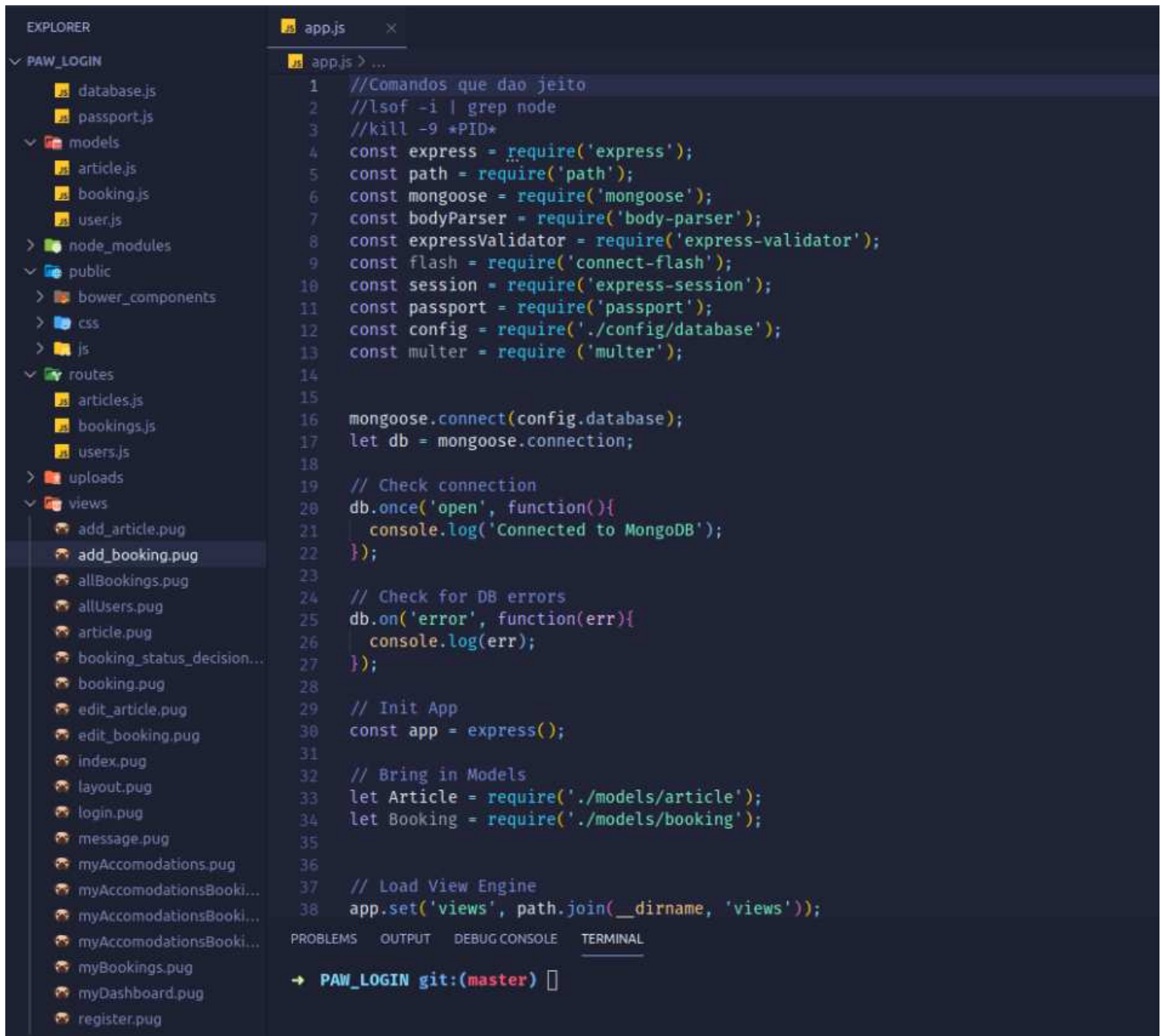
- Registrar/Logar um conta user
- Assegurar estado de autenticacao de user
- Listar Users
- Criar/Listar/Editar/Remover Artigos
- Criar/Listar/Editar/Remover Bookings
- Decidir proposta de Bookings
- Dashboard para clientes
- Dashboard para administradores de Apartamentos
- Consulta de propostas efetuadas de booking
- Consulta de propostas recebidas de booking

Requisitos Não Funcionais

- O sistema deve ser implementado na Linguagem JavaScript
- O sistema deverá comunicar com a base dados MongoDB
- O sistema deve ser executavel em qualquer browser
- O sistema deve implementar Pug client side
- A confiabilidade tem ligação com o tempo médio que o sistema pode vir a falhar, a disponibilidade ou até mesmo a taxa de ocorrência de falhas.
- A métrica da portabilidade é o número de sistema-alvo
- O tamanho pode ser medido em kbytes e número de Chip de RAM
- A velocidade está ligada ao tempo de utilização da tela, ou transações processadas por segundos.

Representacoes Visuais do Desenvolvimento

ManageMyApartment



```
EXPLORER
PAW_LOGIN
  database.js
  passport.js
  models
    article.js
    booking.js
    user.js
  node_modules
  public
  bower_components
  css
  js
  routes
    articles.js
    bookings.js
    users.js
  uploads
  views
    add_article.pug
    add_booking.pug
    allBookings.pug
    allUsers.pug
    article.pug
    booking_status_decision...
    booking.pug
    edit_article.pug
    edit_booking.pug
    index.pug
    layout.pug
    login.pug
    message.pug
    myAccommodations.pug
    myAccommodationsBooki...
    myAccommodationsBooki...
    myAccommodationsBooki...
    myBookings.pug
    myDashboard.pug
    register.pug

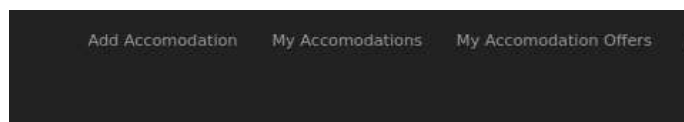
app.js
1 //Comandos que dao jeito
2 //lsof -i | grep node
3 //kill -9 *PID*
4 const express = require('express');
5 const path = require('path');
6 const mongoose = require('mongoose');
7 const bodyParser = require('body-parser');
8 const expressValidator = require('express-validator');
9 const flash = require('connect-flash');
10 const session = require('express-session');
11 const passport = require('passport');
12 const config = require('./config/database');
13 const multer = require('multer');
14
15
16 mongoose.connect(config.database);
17 let db = mongoose.connection;
18
19 // Check connection
20 db.once('open', function(){
21   console.log('Connected to MongoDB');
22 });
23
24 // Check for DB errors
25 db.on('error', function(err){
26   console.log(err);
27 });
28
29 // Init App
30 const app = express();
31
32 // Bring in Models
33 let Article = require('./models/article');
34 let Booking = require('./models/booking');
35
36
37 // Load View Engine
38 app.set('views', path.join(__dirname, 'views'));
39
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
→ PAW_LOGIN git:(master) □
```

Imagem 3 - Screenshot Visual Studio Code no ficheiro App.js

Representacoes Visuais do Desenvolvimento

ManageMyApartment

Imagem 5 - Apresentacao de article pelo seu proprietario



Accomodation: cathy home 1



Owned by Cathy

Region: Azores

Address: Rua do curral

Max Capacity: 10 adults

Price per Stay: 15 €

Features:

Elevator,Balcony,Terrace,HandicapAccessible

Premium Features:

WIFI

AC

Fire Wood

Edit

Delete

ESTG - Booking PandaPanda © 2019

Accomodations

Register

Name:

Email:

Username:

Password:

Confirm:

NIF:

Address:

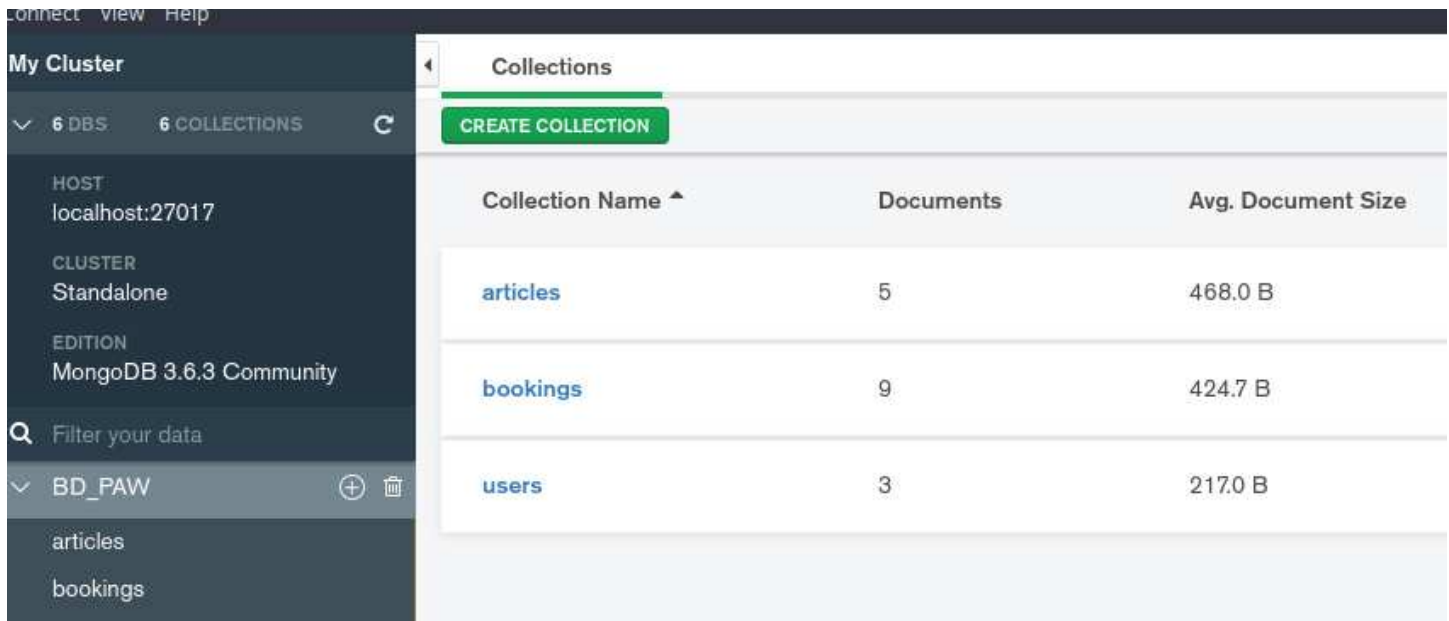
Submit

ESTG - Booking PandaPanda © 2019

Imagem 4 - Formulario web Registo de utilizador

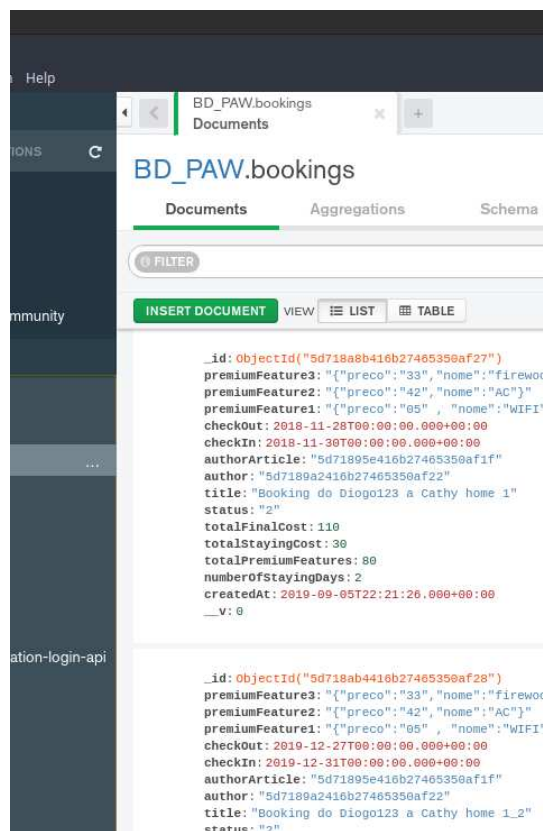
Representações Visuais do Desenvolvimento

ManageMyApartment



Collection Name	Documents	Avg. Document Size
articles	5	468.0 B
bookings	9	424.7 B
users	3	217.0 B

Imagem 6 - MongoDB Compass: Base Dados na Interface Grafica



```
{ "_id": ObjectId("5d718a8b416b27465359af27"),  
  "premiumFeature3": {"preco": "33", "nome": "firewo"},  
  "premiumFeature2": {"preco": "42", "nome": "AC"},  
  "premiumFeature1": {"preco": "05", "nome": "WIFI"},  
  "checkOut": "2018-11-28T00:00:00.000+00:00",  
  "checkIn": "2018-11-30T00:00:00.000+00:00",  
  "authorArticle": "5d71895e416b27465359af1f",  
  "author": "5d7189a2416b27465359af22",  
  "title": "Booking do Diogo123 a Cathy home 1",  
  "status": "2",  
  "totalFinalCost": 110,  
  "totalStayingCost": 30,  
  "totalPremiumFeatures": 80,  
  "numberOfStayingDays": 2,  
  "createdAt": "2019-09-05T22:21:26.000+00:00",  
  "_v": 0 }
```

Imagem 7 - MongoDB Compass: Schema de Bookings

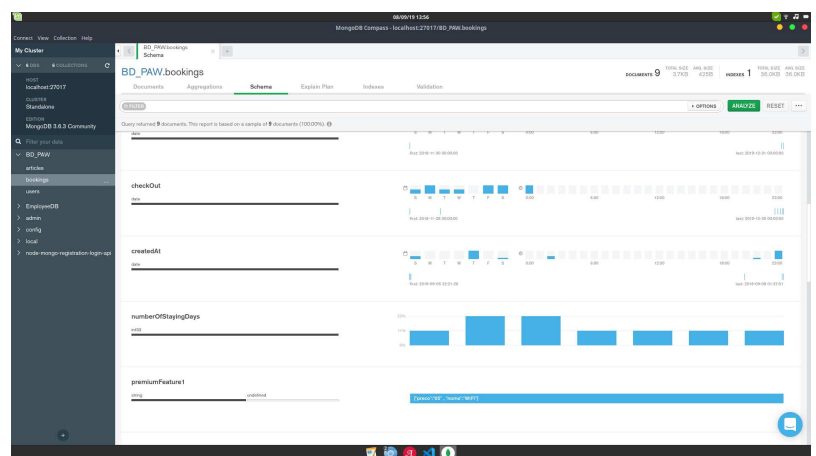


Imagem 8 - MongoDB Compass: Documentos de Booking

Representacoes Visuais do Desenvolvimento

ManageMyApartment

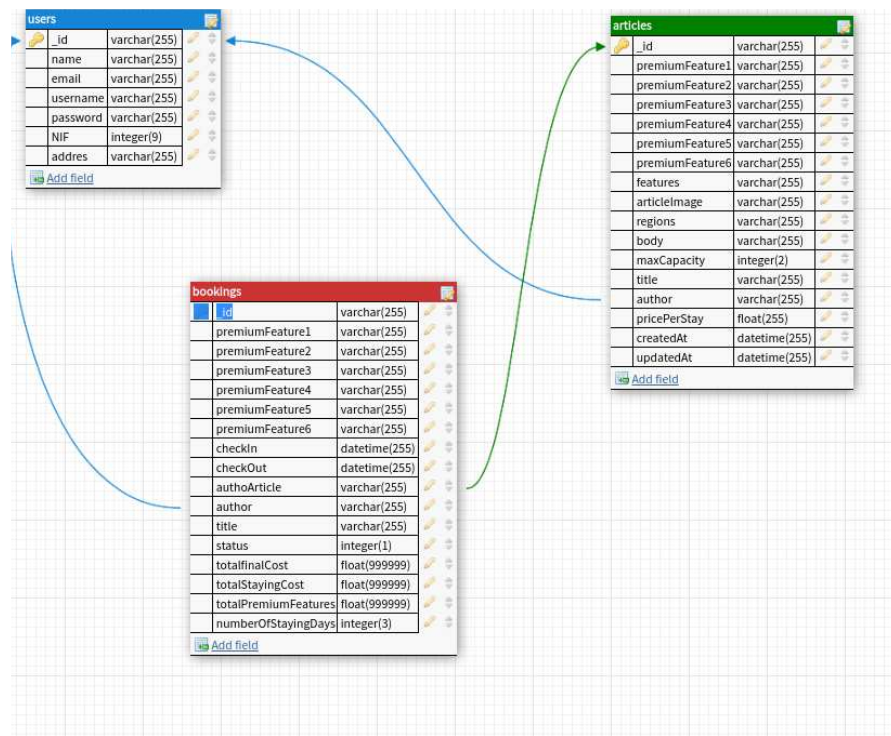


Imagem 9 - Representacao de entidades com respectivos campos e relacoes

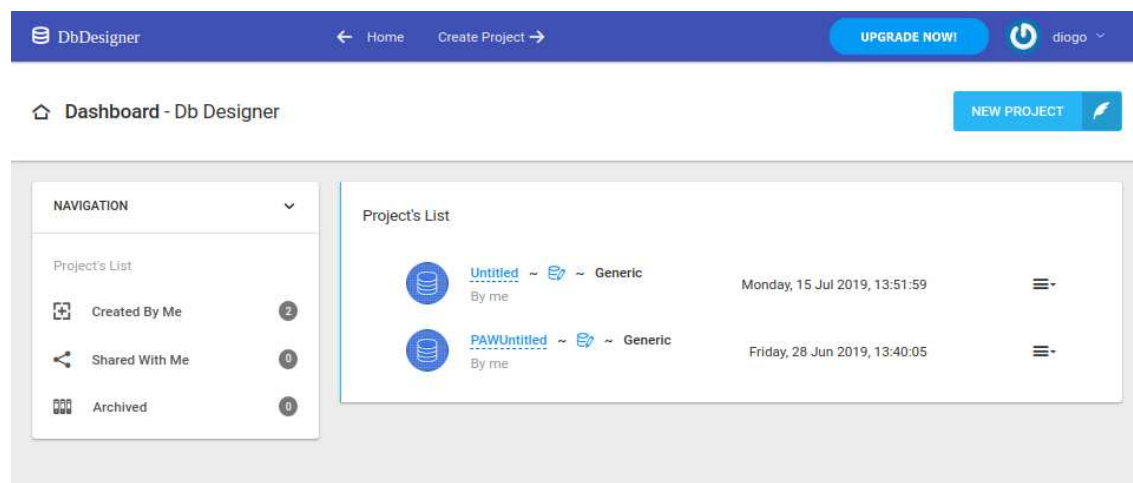


Imagem 10 - BB designer screenshot de ficheiro acima demonstrado

Representacoes Visuais do Desenvolvimento

ManageMyApartment

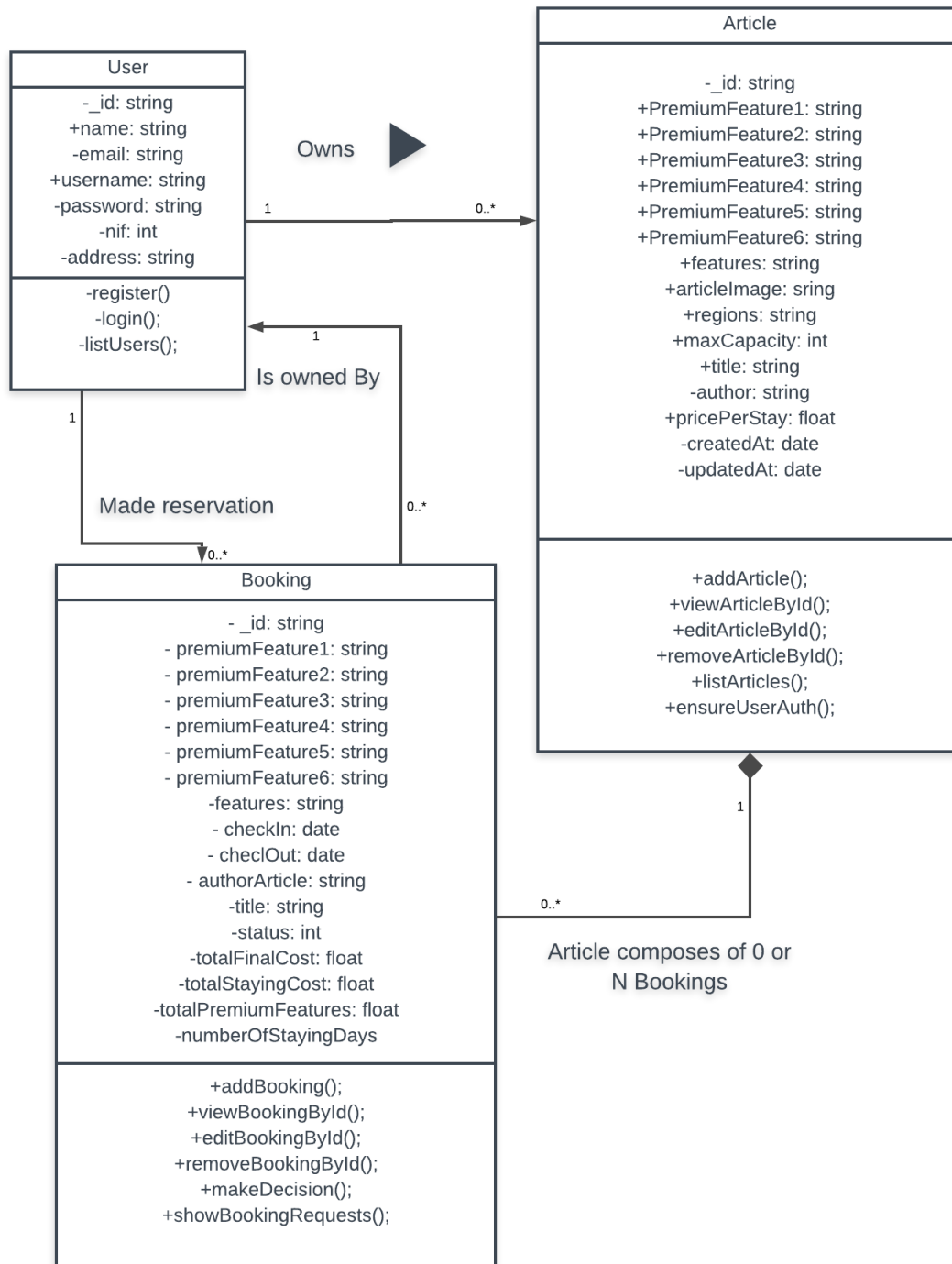


Imagem 11 - Diagrama de classes e respectivas relacoes

Modulos installados que vale a pena mencionar

ManageMyApartment

Antes de mais, o que é um modulo?

Um modulo é uma biblioteca/ficheiro em JavaScript que podemos importar invocando o `require()` do Node

Express é um modulo. Tal como as bibliotecas middleware e base de dados que utilizamos nas nossas aplicacoes Express.

Body-parser foi o modulo que utilizamos com mais frequencia de forma a extrair informacao que desejamos dos pedidos http para uma vareavel que ja podiamos manusear "`req.body`"

Mongoose é um *Object Data Modeling* (ODM) que é uma biblioteca para MongoDB e Node.js. Utilizamos esse modulo para todos os processos em que envolvia guardar ou obter dados armazenados na BD do Projeto "BD_PAW_TP".

Passport serve de middleware de autenticacao para o Node.js. Foi implementado no projeto na parte inicial de Registo de user e posteriormente para Login. Com ele utilizamos uma estrategia de JWT(JSON Web Token) aliado do "bcrypt" para guardar e encryptar a nossa password no banco de dados.

Multer serve para ser possivel carregar multipart/form-data apartir da submissao de formulario. Utilizamos este modulo juntamente com uma estrategia de armazenamento para ser possivel fazer upload e armazenar imagens na criacao de Anuncio

Alguns Conceitos Relevantes

ManageMyApartment

MVC é um acrônimo para para Model-View-Controller. Este é o padrão de arquitetura para desenvolvimento de projetos software. Ele é mais comum em desenvolvimento em Node, C#, Ruby e PHP frameworks.

UML significa Unified Modeling Language. O propósito é visualmente representar sistemas e o seu funcionamento com os seus atores principais, classes, entidades, papéis(funções), de forma a melhor compreender e demonstrar as suas relações/dinâmicas.

GUI traduz Graphical User Interface. Neste projeto foi utilizado MongoDB como Gestor de Base de Dados e para melhor visualização da sua estrutura foi utilizado MongoDB Compass que é o GUI do MongoDB

API tem sentido de Application Program Interface. Este tem um conjunto de rotinas, protocolos e ferramentas para construir aplicações software. Basicamente, uma API especifica como os componentes devem interagir. APIs são também utilizadas quando se programa por componente GUI. O nosso projeto no fundo, foi criar uma API entre o cliente, por exemplo com formulários em Pug o servidor e consequentemente o SGBD MongoDB

SGBD - Sistema de Gestão de Base de Dados. Neste projeto, foi escolhido MongoDB

Considerações Finais

ManageMyApartment

Considero a elaboracao deste projeto tenha sido uma tarefa ardua e complexa, porém muito gratificante apos obter o produto final e todos o conhecimento que fora assimilado nas mais variadas frameworks utilizadas bem como as linguagens em que se desenvolveu cada uma das partes e a consolidacao das respectivas sintaxes, comportamentos, sintaxes e particularidades de cada uma das linguagens/frameworks. Em relacao a maiores dificuldades, penso ter sido a parte de JavaScript e das caracteristicas fracamente tipadas que teem as variaveis o que se transformou numa barreira de evolucao. Apos essa barreira transposta julgo estar muito mais confortavel e agil na compreensao e desenvolvimento em JavaScript e uma maior autonomia na resolucao de problemas.

ManageMyApartment

Para além de todo o Source Code deste projeto, decidi criar um ficheiro txt onde explico, detalhadamente, como iniciei este projeto, que comandos utilizei pelo NodePackageManager para instalar os módulos que fui necessitando e a razão pela qual o utilizei.

Explica também como comecei a implementar MongoDB e o seu GUI, como também como utilizar a ferramenta Git apontar para o repositório criado no serviço de alojamento Github e armazenar o código desenvolvido sem antes de criar rapidamente o ficheiro ".gitignore" da pasta módulos e explica lá o porque de ser fazer isso bem como alguns comandos de terminal que são muito úteis para resolver problemas comuns de portas devido à natureza do nodemon de constante refresh que por vezes tenta aceder PORT que ainda não foi fechado. Finalmente explica como fazer backup da DB ou cópia do MongoDB em Ubuntu para um folder que desejarmos através do Mongodump e a respectiva instalação.

Link repositório Github do projeto + Ficheiro **"README"**

https://github.com/diogozup/PAW_TP_EE.git



```
File Edit View Search Tools Documents Help
#SUMMARY-----
1.1- What does mean restfull
1.2- What tools i used for this project
1.3- Criar lista de tarefas de como comecar 1 projeto de raiz
1.4- START PROJECT RUNNING
1.5- VSCode stuff(atalhos keyboard + codigos conflito ports..)
1.6- Create gitignore file
1.7- Guardarprojetoemrepositorio GIT
-----
1.1- What does mean restfull
REST or RESTful API design (Representational State Transfer) is designed to take advantage of ex
Design was defined by Dr. Roy Fielding in his 2000 doctorate dissertation.
-----
1.2- What tools i used for this project
1- Node with middleware Express.js w/:
  -bootstrap
-----
2- MongoDB for DBstorage and management
-----
1.3- Criar Lista de tarefas de como comecar 1 projeto de raiz
1- Criar pasta com nome projeto
-dentro dela no VSCodeTerminal:
  -npm init -yes
  -create files: (server.js + folders: ["models"+file(db.js)],"controllers","views")
  (meter la codigo de iniciar app em "server")
  -npm i express
  -$sudo npm i -g nodemon
  -nodemon server
  (optional)$export PORT=5000
-----
-Using a BD in MONGODB:
  -(IF NOT INSTALLED MONGO IN PC ALREADY) go to Linux_Term:
  $sudo apt-get install mongodb
  $sudo apt-get update
  $sudo service mongo start
  $mongo || $mongo , cant remember
  $show db || show dbs , cant remember
  -VSCodeTerm: $npm i mongoose --save
  -VSCodeTerm: $npm i body-parser --save //this is very important module for bd usage
  -VSCodeTerm: $mongo //START BD
  -ABRIR MONGO COMPASS e CRIAR BD que vamos USAR
-----
1.4- START PROJECT RUNNING
-open VSCodeProj
  -VSCodeTerm:1: nodemon server
  -VSCodeTerm:2: $mongo ( a DB esta no root folder ja com permissoes: "/data/db")
  ##NOTA:
  #Resposta para permissoes esta no comentario mais votado do StackOverflw a esta
  questao em:
  https://stackoverflow.com/questions/7948789/mongodb-complains-that-there-is-no-data-db-folder
```

```
em VSCodeProj)
CodeTerm:1: $nodemon server
CodeTerm:2: $mongo ( a DB esta no root folder ja com permissoes: "/data/db")
NOTA:
posta para permissoes esta no comentario mais votado do StackOverflw a esta
stao em:
ps://stackoverflow.com/questions/7948789/mongodb-complains-that-there-is-no-data-db-fold
-----
- VSCode stuff(atalhos keyboard + codigos conflito ports..)
se node port ja ocupado:
check ProcessID in VSCodeTerm: $ps aux | grep node
check ProcessID in VSCodeTerm: $lsof -i | grep node
kill ProcessID in VSCodeTerm: $kill -9 12920 (PID = 12920 neste exemplo)
r + shift + 7
omment line/block
r + shift + I
utoFormat Code
-----
- Create gitignore file
o projeto no VSCodeTerm:3:
$git init //inicializar git
$git status //ver os estado
o root do projeto criamos file ".gitignore" e editamos a linha:
i| node_modules/
oltamos no VSCodeTerm:
$git status //revemos os files que temos para fazer commit e reparamos "node_module
$git add //adiciona todos os files
$git commit -m "Our 1st commit"
-----
- Guardarprojetoemrepositorio GIT
epois de todos os pacotes de 1.6 fazemos em VSCodeTerm:
$git status //verificar se os documentos que fizemos commit ja nao aparecem listados
$git log //ver se o commit ta prnto a enviar para repo (ate aqui so temos alterac
nviamos alteracoes feitas no nosso pc para o Repositorio criado:
$git push -u origin master // e metemos credenciais da conta GIT
SE NAO TEMOS REPOSITORIO CRIADO TEMOS QUE CRIAR ASSIM:
r a: https://github.com/
\login (diogozup + pw)
reate Repository:
\https://github.com/new // used in this example RepoName: "NodeExpressJS-MongoDB_CRUD"
ot the commands to "remote add origin" and "push -u origin" //"-"u" significa upstream, d
ommands i got:
)
CoderTerm: $git remote add origin https://github.com/diogozup/NodeExpressJS-MongoDB_CRUD
ter "diogozup" + "pw(hardone)"
lose VSCodeTerm:3 and !!!!!!!DONE !!!!!!!
```